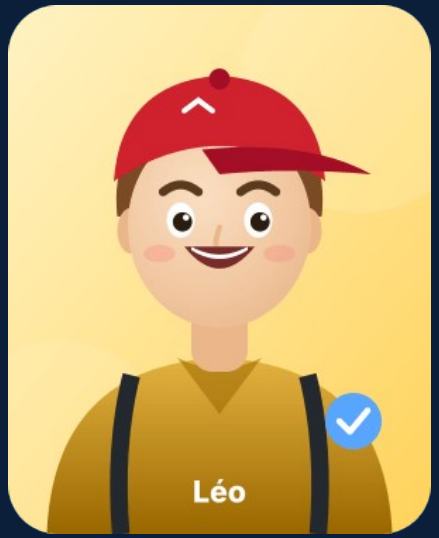
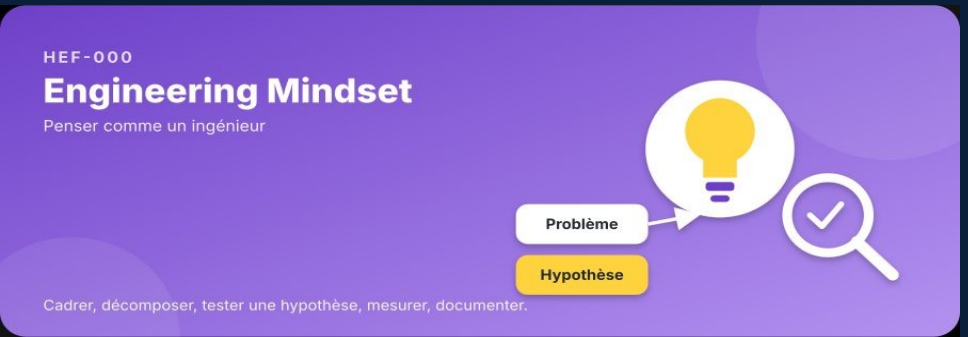
HEF-000 · MISSION 04 / 9

Déboguer avec méthode

Déboguer, ce n'est pas deviner. C'est mener une enquête.





Badge : Enquêteur·rice du code

75 min · débutant

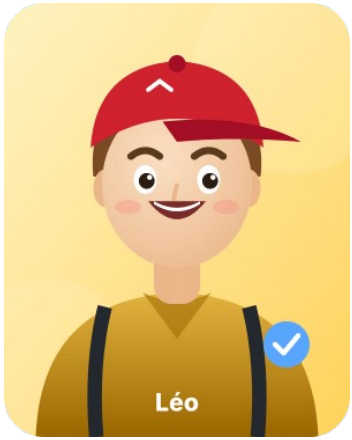
L'histoire

Les utilisateurs ne reçoivent plus les e-mails de réinitialisation de mot de passe.

Kodi, stressé, change trois réglages en même temps, redémarre le serveur, et… ça remarche. Soulagement !

« Qu'est-ce qui a réglé le problème ? demande Sam. — Euh… je ne sais pas. — Alors il reviendra. »

Cette mission t'apprend à trouver la cause, pas seulement à faire disparaître le symptôme.



Ta mission

À la fin, je serai capable de…

1. décrire un bug avec des observations précises ;
2. formuler une hypothèse testable ;
3. concevoir un test qui départage deux explications ;
4. changer une seule chose à la fois ;
5. distinguer la cause immédiate de la cause racine ;

Ta mission

À la fin, je serai capable de…

1. vérifier que la correction a bien fonctionné.

Défi d'échauffement

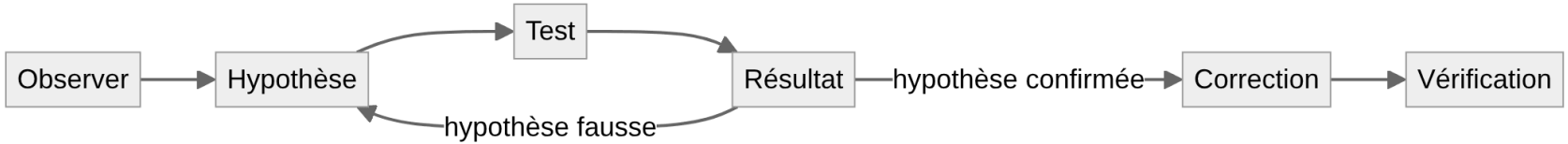
Défi d'échauffement

Ta lampe de chevet ne s'allume plus. Écris **tout ce que tu essaierais**, dans l'ordre.

Ensuite, entoure les essais qui **changent plusieurs choses à la fois**. Pourquoi serait-ce un problème ?

Le concept

La boucle du débogage



Description du schéma : on observe, on formule une hypothèse, on la teste. Si elle est fausse, on revient à l'hypothèse. Si elle est confirmée, on corrige puis on vérifie.

1. Observer avant d'expliquer

Une bonne observation est **précise** et **sans interprétation**.

- ❌ « Le serveur bugue. »

Le concept

1. Observer avant d'expliquer (suite)

- ✅ « Quand un utilisateur connecté envoie le formulaire d'inscription, le navigateur reçoit une erreur HTTP 500. »

2. Une bonne hypothèse

Une hypothèse utile est **précise** et **testable**.

- ❌ « Quelque chose ne va pas côté serveur. » (impossible à tester)
- ✅ « La requête échoue parce que l'application ne peut pas se connecter à la base de données. »

3. Un bon test

Un bon test **départage** : selon le résultat, on garde ou on abandonne une hypothèse.

Le concept

3. Un bon test (suite)

Hypothèse : « la base est injoignable ». Test : tenter une connexion à la base et lire l'erreur.

4. Une seule chose à la fois

Si tu changes 3 choses et que ça marche, tu ne sais pas laquelle a servi. Change **une** variable, observe, recommence.

5. Reproduire et isoler

Un bug que tu peux **provoquer à volonté** est à moitié résolu. **Isoler**, c'est réduire le contexte (moins de code, moins de données) jusqu'à trouver le plus petit cas qui plante.

Le concept

6. Cause immédiate et cause racine

- **Cause immédiate** : ce qui a déclenché l'erreur. « Le serveur d'e-mail a refusé l'envoi. »
- **Cause racine** : pourquoi c'est arrivé. « Le mot de passe du serveur d'e-mail a expiré, et personne n'est alerté. »

Corriger la cause immédiate soulage. Corriger la cause racine **empêche le retour**.

🧠 ***Technique des 5 pourquoi*** *: demande « pourquoi ? » plusieurs fois de suite jusqu'à une cause que tu peux vraiment corriger.*

7. Vérifier

Le concept

7. Vérifier (suite)

Après la correction : refais le scénario qui échouait, vérifie qu'il marche, et vérifie que **rien d'autre n'est cassé**.

Exemple pas à pas

Bug : les e-mails de réinitialisation ne sont plus reçus.

Observation : l'utilisateur clique, voit « e-mail envoyé », rien dans sa boîte, depuis lundi.

Hypothèses :

- H1 : l'application n'envoie pas l'e-mail.
- H2 : l'e-mail est envoyé mais refusé par le service d'envoi.
- H3 : l'e-mail est livré mais classé en indésirable.

Test pour départager H1 et H2 : lire les journaux de l'application à l'heure du clic.

Résultat : on voit « Erreur 535 : authentification refusée par le serveur d'e-mail ». H1 est fausse (l'appli essaie d'envoyer) et H2 est confirmée.

Exemple pas à pas (suite)

Cause racine : les identifiants du service d'e-mail ont expiré lundi.
Correction : renouveler les identifiants, et ajouter une alerte avant expiration.

Vérification : une réinitialisation de test arrive bien, et les autres e-mails (confirmation d'inscription) aussi.

À toi, avec aide

Bug : *« Après avoir terminé un quiz, l'écran affiche "Terminé", mais après actualisation, la progression revient à "Non terminé". »*

1. Écris l'observation précise.
2. Propose 3 hypothèses.
3. Pour chacune, un test.

À toi, avec aide · réponse

Indices progressifs

Indice 1

L'écran affiche « Terminé » avant l'actualisation : qu'est-ce que ça dit sur ce qui est enregistré ou non ?

Indice 2

Hypothèses possibles : l'enregistrement n'est jamais envoyé ; il est envoyé mais refusé ; il est enregistré mais relu au mauvais endroit.

Indice 3

Pour tester, regarde l'onglet réseau du navigateur, puis la base.

À toi, avec aide · réponse

Exemple de réponse

Observation : l'interface affiche « Terminé » juste après le quiz, mais l'état redevient « Non terminé » après rechargement.

H1 : la requête d'enregistrement n'est pas envoyée → test : onglet réseau. H2 : la requête est envoyée mais échoue → test : code de réponse et journaux. H3 : la donnée est enregistrée mais relue d'un ancien champ → test : regarder la base après le quiz.

À toi, en solo

Un site de réservation affiche, pour certains utilisateurs seulement, « Aucune place disponible » alors qu'il en reste.

1. Écris trois observations à demander.
2. Écris deux hypothèses et un test pour chacune.
3. Écris la cause racine possible et une façon de l'empêcher de revenir.

À toi, en solo · réponse

Pistes de correction

Observations : quels utilisateurs ? depuis quand ? sur quel appareil ? Hypothèses : un cache périmé pour certains utilisateurs ; un filtre par région mal configuré. Une bonne réponse dit **comment** vérifier (comparer deux utilisateurs, vider le cache sur un seul, etc.) et propose une protection (test automatique, alerte).

Pièges classiques

Piège	Pourquoi c'est un problème	Comment l'éviter
Essais au hasard	On ne sait pas pourquoi ça marche	Toujours une hypothèse avant un essai
Biais de confirmation	On ne cherche que ce qui confirme notre idée	Chercher aussi ce qui la contredit
Plusieurs changements à la fois	Impossible de savoir lequel a servi	Un changement, un test
Corriger le symptôme	Le bug revient	Chercher la cause racine
S'arrêter à « ça va mieux »	Le bug est peut-être juste masqué	Vérifier avec le scénario exact

Mini-boss

Mini-boss

Un client dit : *« Mon paiement a été débité deux fois. »*

Rédige un **journal de débogage** avec :

1. observation précise ;
2. 3 hypothèses classées par probabilité (avec justification) ;
3. un test par hypothèse ;
4. une cause racine possible ;
5. une vérification après correction ;
6. une mesure pour empêcher la récidive.

Mini-boss · réponse

Pistes

Hypothèses : l'utilisateur a cliqué deux fois ; le réseau a relancé la requête automatiquement ; le fournisseur de paiement a renvoyé une confirmation deux fois. Mesure : rendre l'opération « idempotente » (la même demande ne produit qu'un seul débit).

Quiz éclair

Quiz éclair

1. Une bonne hypothèse est :

A. vague et large
B. précise et testable
C. impossible à contredire

2. Pourquoi changer une seule chose à la fois ?

A. C'est plus rapide
B. Pour savoir ce qui a fait la différence
C. Pour économiser du temps de serveur

3. La cause racine est :

A. ce qui a déclenché l'erreur

Quiz éclair

Quiz éclair (suite)

B. la raison profonde pour laquelle elle est arrivée
C. le nom du fichier en cause

4. 🔁 *Révision mission 03* : « deux événements simultanés » indiquent :

A. une cause prouvée
B. une corrélation à tester
C. une erreur

5. 🔁 *Révision mission 02* : décomposer par processus, c'est :

A. lister les étapes que traverse la demande
B. lister les équipes
C. lister les fichiers

Quiz éclair · réponse

Réponses

B. 2. **B**. 3. **B**. 4. **B**. 5. **A**.

Je réfléchis

1. Reprends ton défi de la lampe. Que ferais-tu autrement ?
2. Quelle est la dernière fois où tu as « réparé » quelque chose sans savoir pourquoi ça marchait ?

À retenir

À retenir

1. Observer d'abord, expliquer ensuite.
2. Une hypothèse doit être testable.
3. Un changement à la fois.
4. Chercher la cause racine, pas seulement le symptôme.
5. Toujours vérifier après la correction.

MISSION ACCOMPLIE

+100 XP

Badge débloqué : Enquêteur·rice du code

Prochaine mission : Documenter et expliquer

